A Survey of Euler Diagrams

Peter Rodgers

University of Kent, UK
p.j.rodgers@kent.ac.uk

Abstract

Euler diagrams visually represent containment, intersection and exclusion using closed curves. They first appeared several hundred years ago, however, there has been a resurgence in Euler diagram research in the twenty-first century. This was initially driven by their use in visual languages, where they can be used to represent logical expressions diagrammatically. This work lead to the requirement to automatically generate Euler diagrams from an abstract description. The ability to generate diagrams has accelerated their use in information visualization, both in the standard case where multiple grouping of data items inside curves is required and in the area-proportional case where the area of curve intersections is important. As a result, examining the usability of Euler diagrams has become an important aspect of this research. Usability has been investigated by empirical studies, but much research has concentrated on *wellformedness*, which concerns how curves and other features of the diagram interrelate. This work has revealed the drawability of Euler diagrams under various wellformedness properties and has developed embedding methods that meet these properties.

Euler diagram research surveyed in this paper includes theoretical results, generation techniques, transformation methods and the development of automated reasoning systems for Euler diagrams. It also overviews application areas and the ways in which Euler diagrams have been extended.

Key words: Euler diagrams

1. Introduction

The embedding, application and theoretical underpinning of Euler diagrams has been the focus of much recent research effort. This paper surveys the state-of-the-art in the area.

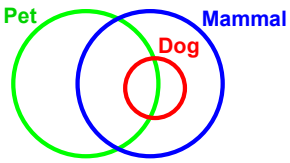

Figure 1: An example Euler diagram.

Euler diagrams are collections of labelled, closed curves. They visually represent containment, intersection and exclusion. An example is shown in Figure 1. This diagram can be interpreted as showing that all dogs are mammals, some mammals are pets and some dogs are pets.

The history of Euler diagrams is explored by Baron [4]. She notes that Leibniz produced similar diagrams before Euler, however, much of this work was unpublished. She also observes even earlier Euler-like diagrams by Ramon Lull in the 13th Century. However, logical reasoning using labelled closed curves was first popularized by Leonhard Euler [32] in his 'Lettres à une Princesse d'Allemagne'. These letters are based on his lessons in physics and philosophy between 1760 and 1762 for Princesse d'Anhalt-Dessau, niece of King Frederick II of Prussia. See Figure 2 for an example of his drawings.

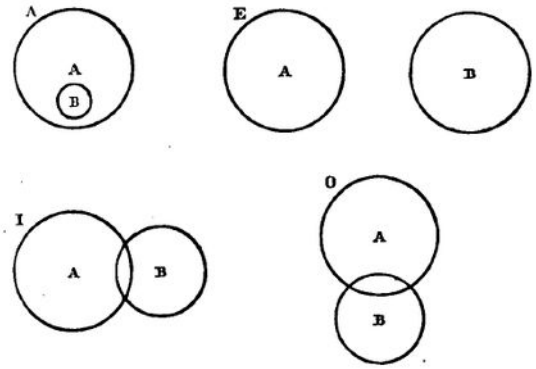

Figure 2: From Euler's original drawings [32].

Alternative names for Euler diagrams include 'Euler circles'. They can also be incorrectly called Venn diagrams. Venn diagrams require all possible curve intersections to be present, so can be seen as a subset of Euler diagrams, that is, every Venn diagram is an Euler diagram, but not every Euler diagram is a Venn diagram. Venn diagrams were introduced by John Venn a hundred years after Euler [117] and shaded empty regions, a strategy that also appears in some modern Euler diagram syntax. Whilst Venn diagrams quickly become convoluted and difficult to

interpret as more curves are added, there are constructions that maintain wellformed properties of the diagrams (wellformedness is discussed in Section 4). There is some evidence to support the use of Euler diagrams over Venn diagrams for deductive reasoning [91]. See Ruskey's excellent survey on Venn diagrams [90] for more details about research in that area.

Figure 3: An illustration of free rides.

Euler diagrams are considered to be an effective means of visualizing containment, intersection and exclusion. The use of Euler diagrams as a mechanism to group items is supported by the preattentive processing concept of closure [116]. Euler diagrams are also considered to aid inference, using the notion of a 'free-ride' [5], where adding a curve can allow the deduction of information not present in either the original diagram or added curve. This can be seen in Figure 3, which is based on an example in [74]. In the leftmost diagram, Q is a subset of P. In the rightmost diagram, R has been added as a subset of Q, which means that R can also be seen to be a subset of P for free. Free-rides, along with the view that many Euler diagram notations are 'well matched to meaning' [49] supports their use in reasoning systems, such as those given in Section 7. Apart from logic and reasoning, Euler diagrams are used in various contexts from Medicine to Software Engineering. Section 2 gives an overview of application areas.

There are alternatives to Euler diagrams. They include LineSets [2], which group items by drawing a line which connects all the items in a set. There is potential for confusion if the user interprets an ordering to the line that is not present or if a line follows a path with many bends. Hypergraphs can also be used for grouping items if drawn in the subset standard [8], however, this method can lead to unwanted, empty, set intersections being present and relies on the notion that items are already laid out, problems that also occur with Bubble Sets [27]. Alternatives for use in logic include Veitch diagrams and Karnaugh maps [13], however these take a rectilinear approach to visualizing intersection, which is not always effective from a usability perspective. Area-proportional Euler diagrams aim to ensure that the regions are of a desired area. This is similar to cartograms [30], where territories are distorted so that their area is of a desired value, representing a quantity associated with the territory (for example, population). However, the construction methods do not consider any closed curves that might surround a number of territories, so using cartograms in place of area-proportional Euler diagrams would mean that the visual containment of a set of regions by a single curve is lost.

The rest of this paper is organized as follows: Section 2 motivates the study of Euler diagrams by exploring the variety of application areas where they are applied; Section 3 gives a definition of Euler diagrams and examines different abstractions and representations of Euler diagrams; Section 4 explores the concept of wellformedness, a crucial component of Euler diagram visualization; Section 5 looks at automated techniques for embedding Euler diagrams; Section 6 provides an outline of systems that transform Euler diagrams; Section 7 gives an overview of the reasoning systems that are based on Euler diagrams; Section 8 examines research in the case where regions are required to be a specified area; Section 9 looks at cases where Euler diagrams have been extended. We end each of these sections by outlining open research questions. Finally, Section 10 concludes.

2. Application Areas

The purpose of this section is to give a flavour of the applications that make use of Euler diagrams. It is divided into common application areas. There is no attempt to be complete, the intention is to explore the types of Euler diagram used in the real world to motivate the research discussed in later sections of this paper. It should be noted here that Wilkinson [125] conducted an informal review of articles from the 2009 volumes of Science, Nature, and online affiliated journals and found 72 Venn or Euler diagrams.

2.1. Medical Data

The results of medical studies are often visualized as three curve area-proportional diagrams. The goal of such diagrams is to visually communicate scientific results. For example, the three circle Euler diagram from a widely cited study [95] shows the intersections of physician-diagnosed asthma, chronic bronchitis, and emphysema within patients with obstructive lung disease, see Figure 4. Euler diagrams have become a widely used technique for this sort of data and there is an interesting journal editorial discussing their use in this field [119]. A similar approach visualizes major enrollment groups in the World Trade Center Health Registry [33]. Euler diagrams can be used to communicate other types of medical information and a diagram to aid diagnosis is shown in Figure 5. This is called the 'Anorectal complaints algorithm. Venn diagram' and demonstrates the common misuse of the term 'Venn diagram' where 'Euler diagram' should be applied.

In this section we can also include a study into Bayesian inference [69], using examples taken from cancer diagnosis. It tested five different visualizations with an Euler diagram basis, and one other that was not Euler diagram based. They conclude that their studies indictate that "... simply adding a visualization to a textual Bayesian problem is of little help, even when the text refers to the visualization,

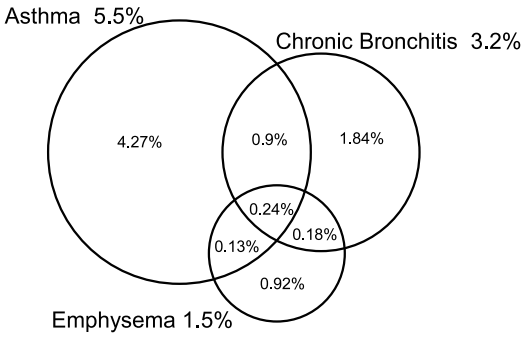

Figure 4: The proportional Venn diagram of obstructive lung disease [95].

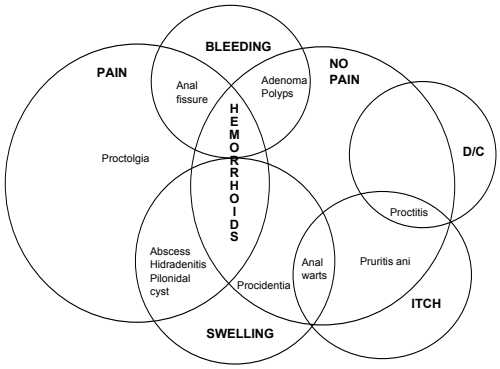

Figure 5: Anorectal complaints [52].

but suggests that visualizations are more effective when the text is given without numerical values."

2.2. Biosciences

The vast quantity of data being produced by automated biological experiments has meant that experts in this area have looked for alternative visualization techniques to help make sense of it, and various software tools based on Euler diagrams have been developed. VennMaster [66] is a tool for visualizing the Gene Ontology database via the GoMiner software [126], see Figure 6. VennMaster is reasonably stable in terms of software and there are a significant number of papers that cite this system, indicating some take up. VennDiagram [20] is a package for the *R* statistical software system, aimed at bioinformatics. This visualizes Venn diagrams with up to four sets and Euler diagrams with up to three sets, see Figure 7. As the system is more recent, it is not yet clear how much the system is being used in its intended application area.

BioVenn [61] is also aimed at biological data, but restricted to drawing area-proportional circular Venn-3 diagrams. Whilst relatively simple, the interface is easy to use and the output is configurable. As a result there are a significant number of research papers that use BioVenn to visualize their data. The PatternLab proteomic data analyzer [16] is restricted to area-proportional two and three circle Venn diagrams, as part of a larger system for data analysis. As it is still a beta release at time of writing, the impact of the system in the application area is not yet clear. GeneVenn [77] links to NCBI's Entrez Nucleotide database, and visualizes interlinking sets of genes using a fixed area circular Venn-3 diagram. There are a surprisingly large number of papers that use the resultant visualizations, considering the simplicity of the results.

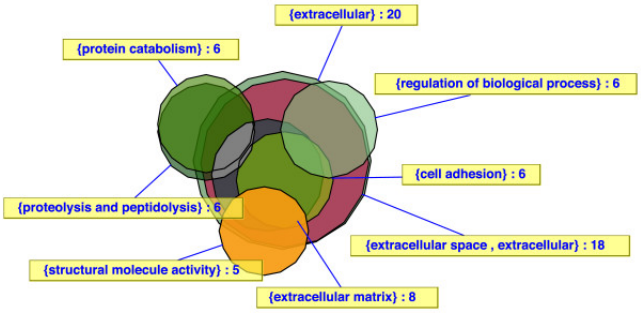

Figure 6: An approximate area-proportional diagram of gene sets derived from a microarray experiment drawn with VennMaster [66].

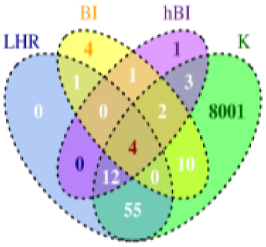

Figure 7: A Venn-4 Diagram visualizing the performance of algorithms proposed to predict bimodal genes drawn with VennDiagram, from [1].

Much of the biosciences use of Euler diagrams described above relates to gene ontologies. In the next subsection we go on to examine other applications that also make use of Euler diagrams to visualize ontologies.

2.3. Classification

In this section we look at application areas that visualize classification data using Euler diagrams. The motivation for using Euler diagrams with this kind of data is that it is possible to surround items with more than one curve, indicating multiple class membership, information that is difficult to communicate with a more traditional tree-based visualization.

There are some interesting hand drawn examples of classification using Euler diagrams. For instance, the Wikipedia diagram of supernational bodies in Figure 8 could be considered an effective way of communicating the membership of various international organizations. In fact, there are a number of similar data sets in Wikipedia that are visualized using Euler diagrams, indicating, to some extent, that Euler diagrams can be an effective visualization method for the general public (because it can be deduced that if they were not effective, the actions of crowdsourcing would remove them from the relevant Wikipedia

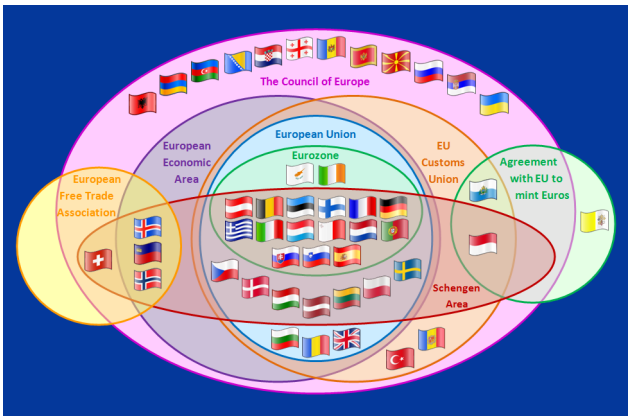

Figure 8: European supernational bodies [124].

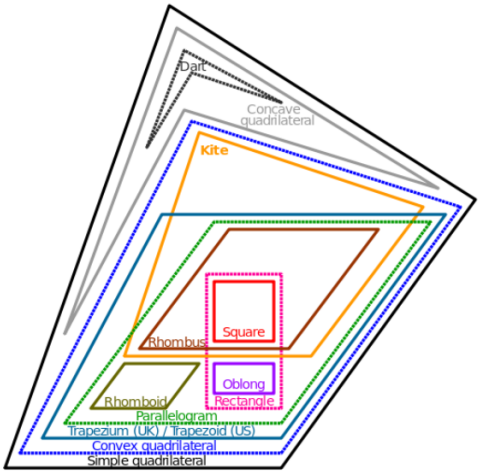

Figure 9: Types of quadrilateral [123].

pages). Visualizing quadrilateral types with an Euler diagram where the shapes of curves correspond to the labels is clever, but may not be the most user friendly diagram, see Figure 9. Other work includes prototype systems to automatically generate methods for displaying information about films, see Figure 10, as well as a suggestion that Euler diagrams would be useful for generating queries in library systems [118].

The FunEuler software [10] lets users create their own Euler diagrams for file management and other applications. It builds upon the VennFS system [29]. A key advantage of Euler diagrams here is that a file can be in several classifications, unlike traditional tree-based file systems, where a file can be placed in only one classification (directory). In addition, Euler diagrams have been used to visualize ontologies in the semantic web [53]. Ontologies hold more knowledge about concepts than their classification including more complex information about their properties and relations. Hence, the use of concept diagrams, an Euler diagram based reasoning system (see Section 7) [60]. Here the goal is to use a visual language based on Euler diagrams to reason about the relationships

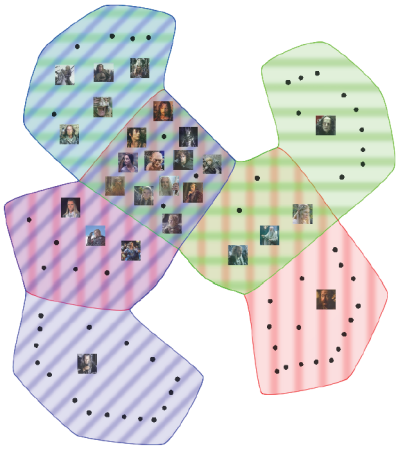

Figure 10: Example output from [94] showing information from a film database. There are three curves, each representing one of the films in the "Lord of the Rings" trilogy. Each item inside the curves represents an actor.

between concepts in ontologies.

2.4. Miscellaneous Applications

Other significant application areas include the use of Euler diagrams as an education tool. This was Euler's original motivation, as he used his diagrams to communicate mathematical ideas, see Section 1. In the present day, teachers typically use Venn and Euler diagrams to explain logical concepts in schools, such as intersection, union and disjointness. Other mathematical areas, such as statistics can also benefit [62] where various Euler diagrams of up to 4 curves are used to visualize multiple regression.

Constraint diagrams are aimed at providing a diagrammatic alternative to symbolic constraint languages for software engineering, such as OCL [75]. This work is motivated by the diagrammatic nature of much software engineering notation, and Engineers' perceived resistance to symbolic logic. This use of Euler diagrams is discussed in more detail in Section 7 because the diagrams are used not only for the visualization of data but are also manipulated with sophisticated visual reasoning systems.

There is a strand of humour that uses Euler diagrams. See, for example, the website thisisindexed.com which regularly posts frivolous Euler diagrams [50].

2.5. Application Open Research Questions

The two main areas for further research in applying Euler diagrams are (1) to investigate how to visualize Euler diagrams so as to display the data most effectively; and (2) the provision of tools so that users can apply Euler diagrams in an accessible and user friendly way. In terms of (1) a few studies have been performed, see Section 4. However, these are limited in scope and, are to some extent contradictory. Important visualization questions remain: what are the best shapes for the curves in a Euler diagram? What colours, shading and labelling methods are

most effective? How do we best display a diagram when data changes? These questions have been addressed in fields such as cartography and graph drawing, which may inspire similar work in relation to Euler diagrams.

The work in relation to (2) concerns the crucial factors of: user interface; integration with data sources; and the automatic generation of Euler diagrams (see Section 5). Various systems may succeed wholly or partially at one, or more rarely two, of these factors, but none provide a complete solution to all. The next evolution in Euler diagram visualization tools surely must be to develop comprehensive solutions to all three factors for particular application areas.

3. Defining Euler Diagrams and Representation

3.1. Basic Definitions

The definition of an Euler diagram varies throughout the literature. Here we take an informal, but fairly typical approach. For a more formal treatment see, for example [103].

An *Euler diagram* can be defined as a finite set of labelled, closed curves.

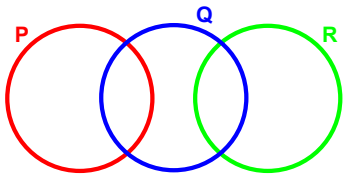

Figure 11: A three curve Euler diagram, $\emptyset$ P Q R PQ QR.

Figure 11 shows an Euler diagram with three curves labelled P, Q and R. An Euler diagram satisfying our definition could also contain duplicate curve labels, non-simple curves and other less desirable features (see Figure 15). If further restrictions on the diagram are required, then the diagram definition can be tightened, for example, unique curve labels can be enforced by defining an injective labelling function, or simple curves can be enforced by using Jordan curves. Occasionally, as an alternative to curves, diagrams are defined using regions in the plane [67], however, as the border of each region can be defined with a curve, this does not differ significantly from the approach given here.

In general, because curve labels may be repeated in the diagram, it is necessary to consider the set of all curves that have the same label; we call this set a *contour*. The Euler diagram in Figure 11 has 3 curves, each forming a contour, but the diagram in Figure 15c has four curves, but three contours, as two curves share the same label R, so forming one contour.

The closed curves in a diagram partition of the plane into *minimal regions*, where each minimal region is a connected component of the plane inside a set of curves (and outside the others).

A *zone* (sometimes called 'concrete zone') is the set of minimal regions that are inside a given set of curves (and outside the other curves). The zone outside all curves (sometimes called the 'outside' or 'empty' zone) is always present. Zones form a key concept in Euler diagrams, as achieving the correct set of zones is the goal of the embedding process. The Euler diagram in Figure 11 has six zones, each formed from a single minimum region. However, the diagram in Figure 15e has eight zones, but nine minimal regions, as the zone contained in only the curve labelled Q is formed from two minimal regions.

3.2. Abstract Description

It is useful to have a way to talk about the essential structure of an Euler diagram without needing to refer to a concrete embedding of curves. An abstract description has two main uses: firstly, it can describe the zones that appear in a given diagram; secondly, it can describe a diagram that has not yet been created, for instance, it can be the input into an automatic embedding mechanism, specifying the zones that should be in the final diagram.

An *abstract zone* is a set of labels. If the abstract zone relates to a Euler diagram that currently exists, then it should correspond to a zone in the diagram and each label should correspond to a contour in the diagram. Hence, the labels must be exactly those that the zone is inside. Where there is no ambiguity, an abstract zone is often just called a zone.

An *abstract description* consists of a set of abstract zones. It must include the abstract zone that contains no curve labels (the empty set), which corresponds to the outside zone that is always present when an Euler diagram is drawn in the plane. When describing an existing Euler diagram, an abstract description should contain exactly the abstract zones that correspond to the zones in the diagram. For example, the abstract description for the Euler diagram in Figure 11, is $\{\emptyset, \{P\}, \{Q\}, \{R\}, \{P, Q\}, \{Q, R\}\}$. The set notation can get in the way of understanding, and so often the the notation is abused by writing this set as $\emptyset$ P Q R PQ QR.

Alternative methods of defining abstractions to that given here have also been proposed, for instance those that explicitly state the sets both inside and outside each zone [35]. There is also alternative terminology, for instance the term 'diagram description' can be used [101] which is synonymous with the notion of an abstract description.

We note that the definition of abstract description given here would suffice for the definition of an undirected hypergraph, and indeed, as noted in Section 1, hypergraph visualizations are an alternative to Euler diagrams. This identity of representations means that many theoretical concepts and algorithms relating to hypergraphs can be applied to Euler diagrams. One such result is that, because hypergraph isomorphism can be solved in exponential time (on the number of vertices) [68], it follows that Euler diagram isomorphism is exponential (on the number

of labels) [100]. This work also gave a way to count the number of isomorphically distinct diagrams for a number of labels, after noting an analogy with switching theory.

3.3. Representations

There are a number of graphs and other representations that are used to describe Euler diagrams. They have differing levels of abstraction, some describing only data (for example, abstract descriptions, discussed in the previous subsection), others have some notion of zone adjacency, and yet others fully define the concrete representation of the curves. One reason for this proliferation of representations is due to the stages required for embedding a concrete diagram from an abstract description, discussed in Section 5. Another reason is that some operations on Euler diagrams, such as transformations, are difficult to explain and reason about with concrete representations that fully define the curves, but are easier when less precision about the embedding is required. Theoretical results on some of the graphs associated with Euler diagrams can be found in [88]. Table 1 summaries the representations discussed in this subsection.

Table 1: Euler diagram representations, ordered with the most concrete at the top and the most abstract at the bottom.

Representation	**Typical Characterization**
Image	Pixels/Ink
Concrete diagram	Mathematical definition of curves
Euler graph	Graph
Concrete dual	Graph
Superdual	Graph
Abstract Description	Set

A *concrete representation* is one that defines the points on the plane for each closed curve. This can be as simple as defining a centre point and radius if the diagram includes circles, alternatively, a continuous function where the end and start points are the same can be used. The *image* of a diagram is not the same as a set of curve definitions, as there are many possible functions that can create the same image. For example, images with apparently simple curves can, in fact, be formed from non-simple curves that overlap themselves. In Euler diagram research, the relationship between images and concrete representations usually follows a pragmatic approach where, if not explicitly stated, the interpretation is that the image has been produced by the least complex concrete representation.

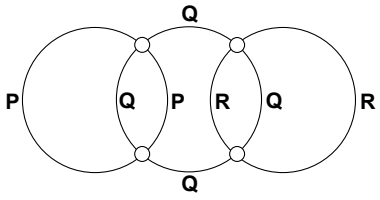

Figure 12: A Euler graph of the Euler diagram shown in Figure 11.

A concept close to the concrete representation is the *Euler graph* of a diagram. This was first defined in [22], drawing on similar concepts for Venn diagrams [90]. An Euler graph is shown in Figure 12. It is the Euler graph of the Euler diagram given in Figure 11. An Euler graph of an Euler diagram can be formed by placing a vertex at each point of intersection and connecting these vertices by undirected edges that follow the curve segments between them. Concurrent curve segments are represented by a single edge. The edges are labelled by the set of curves from which they are formed. An additional vertex is added to any curve which does not intersect with any other curves and that curve is then represented by a self-sourcing edge in the Euler graph. An Euler graph is a plane graph, where the faces of the plane graph correspond to the zones in the diagram.

There is some potential for confusion, as *Eulerian graphs* are sometimes called Euler graphs in graph theory. These are connected graphs where all nodes have even degree. Euler graphs are Eulerian for Euler diagrams that are connected and do not have concurrency (see Section 4 for more detail on concurrency). However, in the general case, the Euler graphs that are discussed in Euler diagram research are not Eulerian graphs.

The reproducibility of an Euler diagram from its Euler graph depends on the information encoded in the edges of the Euler graph. If the edges precisely follow the route of the curve segments that they correspond to, then the Euler diagram is completely reproducible. However, some abstraction is still possible if the edges do not encode this level of detail. What needs to be encoded at a minimum is the ordering of edges around a node. With this, an Euler diagram may be embedded with the same zones, zone adjacency, and wellformedness (see Section 4) as the original, but not necessarily with the same curve shape.

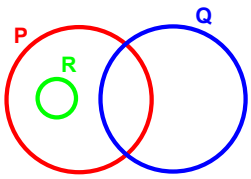

Figure 13: A disconnected Euler diagram, $\emptyset$ P Q PQ PR.

The Euler graph can be used to define the connectedness of an Euler diagram. A *connected* Euler diagram is one with a connected Euler graph, that is, where there is a path between any pair of vertices. If an Euler diagram has a disconnected Euler graph, then the Euler diagram is disconnected. For example, Figure 11 is connected, whereas Figure 13 is disconnected. Alternative terminology includes *atomic* for connected and *nested* for disconnected [42].

Another key graph abstraction is the concept of the *concrete dual* of an Euler diagram introduced in [41] as the *plane dual*, which is similar to the pre-existing concept of the dual of a Venn diagram [90]. Given an Euler diagram,

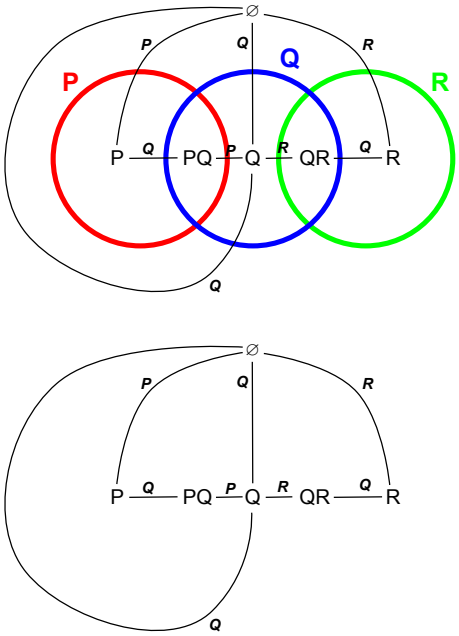

Figure 14: A concrete dual of the Euler diagram shown in Figure 11. Above, the concrete dual is superimposed on the Euler diagram, below is it shown on its own.

its concrete dual is a graph where each vertex corresponds to a zone and each edge connects adjacent zones. The vertices are labelled with the curves that the zone is contained in and the edges can be labelled with the set of curve labels on the curve segment that separates the two zones, although this edge label can be derived from the vertex labels. Figure 14 shows the concrete dual of the Euler diagram of Figure 11. Another way of defining the concrete dual of an Euler diagram is as the dual graph of the Euler graph of the diagram. Hence, a concrete dual can be formed by finding the Euler graph, then finding its dual graph.

The *superdual* is a graph that is formed from the abstract description, a concept introduced in [41]. There is a vertex for each abstract zone in the abstract description, each labelled with the abstract zone to appear. Edges are added where there is one difference in the label sets. That is, two connected vertices have the same label, except for one element. This difference can be used to label the edge between them. Although the concrete dual in Figure 14 is also the superdual of the diagram's abstract description, this is not necessarily so, as can be seen in Figure 17a, in addition, the graph is not necessarily planar. The superdual plays an important part in several embedding methods, see Section 5.

The are other representations which often combine or extend those described here. As they are used in specific techniques for embedding diagrams, they will be discussed in later sections.

3.4. Representation Open Research Questions

The current non-concrete representations largely rely on graph based models. It is certainly possible to derive alternative representations that, for instance, rely on a triangle mesh or other geometric structures. Other topological structures such as manifolds may permit the abstract representation of Euler diagrams. It might be a useful intellectual exercise to define new representations, but without having obvious benefit, perhaps in terms of enabling a new embedding method or allowing more reasoning about a diagram, such alternative representations would remain curiosities.

Perhaps more practically useful would be to encode more subtle detail about geometry in the representation. At present the routing of a curve is either preserved entirely or not encoded at all, bar some notion of edge ordering around a vertex, where the edge represents a curve segment. It may be possible to encode less direct information about the curvature of curves, such as: specifying the direction in which it bends; giving a region in which the curve can be drawn; or defining an allowable rate of turn for the curve. This might allow important geometric aspects of curves to be defined, without fully prescribing their complete paths.

4. Wellformedness Properties

Introduced in [41] and formalized in [108] the concept of a wellformedness property (sometimes called a 'wellformedness condition') relates to relationships between curves and regions in the diagram. Typically, Euler diagram research papers will make explicit or implicit reference to these properties, as various embeddings and transformations are only possible when some properties are present. Wellformedness can be broken in a number of ways. See Figure 15 for examples of each:

1. **n-points** three or more curves cross at the same point (often called 'triple points', especially when three curves meet).
2. **concurrency** two or more curve segments are concurrent.
3. **duplicate curve labels** two or more curves have the same label.
4. **non-simple curves** a curve self intersects.
5. **disconnected zone** a zone consists of more than one minimal region.
6. **brushing points** two or more curves meet at a point but do not cross.

What counts as a wellformedness property is not easily defined, and many other features of diagrams can also be legitimately called wellformedness properties. One such is that of restricting the shape of curves in some way, for example, a property could be defined that ensured that all curves were circles (or more generally ellipses), or that all curves were rectilinear. Another is that the zone areas

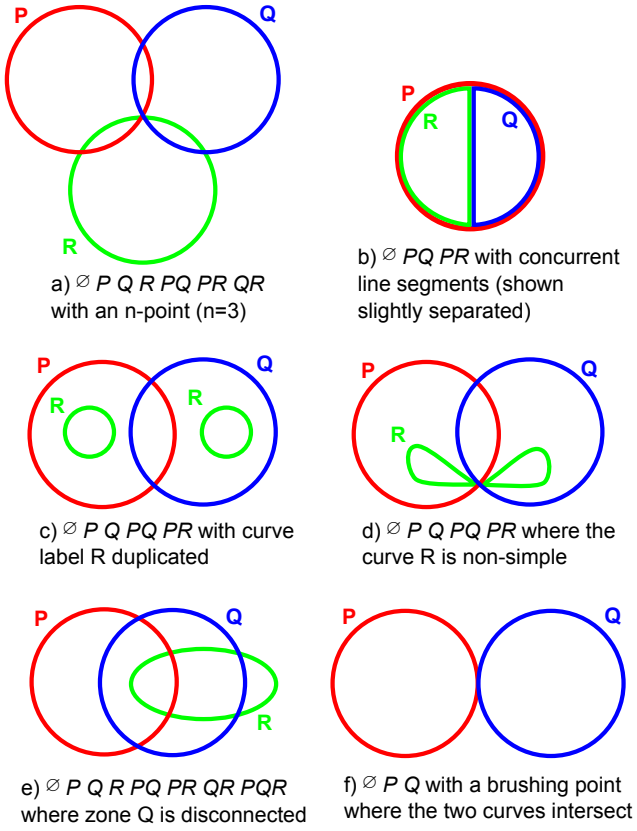

Figure 15: Examples of broken wellformedness properties.

have specific values, or are within certain ranges. These are not included in the above list because, firstly, there are many different interpretations of these properties, and secondly, they are areas of study in their own right, as we shall see in later sections of this survey.

A diagram that adheres to all relevant wellformedness properties is termed *wellformed*. Although more relaxed requirements for wellformedness properties often means that more data sets can be dealt with. However, diagrams adhering to more wellformedness properties are considered to be more easily interpretable than non-wellformed diagrams, leading to a tension between generality and usability. However, many data sets in the real world cannot be represented with wellformed Euler diagrams, so restricting a system to only wellformed diagrams may be of low utility.

The non-wellformed case is often difficult to address and so a partial solution that deals with the wellformed case can then later lead to a more general solution for all data sets. For example, this is the research trajectory taken by the generation problem, see Section 5. It should be noted that lifting the brushing point restriction can extend some theoretical results, as it means that disconnected diagrams can be connected by simply pushing the disconnected components together so that they touch [22]. This allows disconnected diagrams to be reasoned about as if they were connected diagrams. It should also be noted that, for non-wellformed diagrams, it is possible to count the number of times wellformedness is broken, for instance the number of concurrent line segments or the number of n-points can be found. This means that it is possible to describe one embedding of a diagram to be more wellformed than another embedding.

These wellformedness properties have been studied in relation to their usability. Studies on diagrams in a logic context (including empty zones shown with shading) concluded that brushing points and triple points were the properties that most adversely impacted on understanding and that that concurrency can actually aid understanding [39]. Contradictory results were found in two other studies [86] in the context of information visualization, where zones had data items in them. Here, diagrams with concurrency or disconnected zones were shown to be significantly less usable than diagrams with other properties. Diagrams with brushing points were not significantly worse than diagrams without them and diagrams with non-simple curves were less preferred by users than diagrams with other properties. These inconsistencies between the studies may be down to the different application areas studied, and the different types of diagram that were presented to users, or simply due to issues with the methodology that was applied. However, as things stand, without further investigation, reliable guidelines regarding the priority of different wellformedness properties cannot be formed.

Attempts to measure 'clutter' in Euler diagrams have been formulated [63] and a study into more general layout criteria for Euler diagrams, rather than their wellformedness properties has been conducted [7]. All three criteria explored: smoothness of curve; equality of zone area; and curve segment separation were shown to have an impact on user understanding. Finally, in terms of studies on Euler diagrams, an experiment indicates that there is no evidence that diagram rotation has an impact on understanding [9], so validating the approach of within subject studies, which counter the learning effect by rotating diagrams when presenting duplicates to empirical subjects.

4.1. Wellformedness Open Research Questions

Wellformedness properties have been considered to be potentially useful from an usability perspective, but are they? The limited number of studies so far seem to indicate that not all properties are important for all application areas, however more empirical work is needed to confirm and expand these results. More fundamentally, it is possible to question why this set of properties has been chosen. They have been identified in an ad-hoc manner, and so other aspects of the curves and regions might also be included as wellformedness properties. This would be greatly aided by a systematic method for classifying wellformedness conditions, work that has not received much attention.

5. Embedding Methods and Drawability

One of the main subjects of research in recent years has been to examine methods to find a suitable Euler diagram from an abstract data set. The *Euler diagram generation problem* can be stated as:

> Given an abstract description, embed an Euler diagram with only the concrete zones that correspond to the given abstract zones.

This is analogous to the *graph drawing problem*, where the goal is to lay out a graph for better understanding [6]. Manual creation of Euler diagrams, using image editing software is an alternative, and the SketchSet [120] sketch input system has been developed for Euler diagrams in an attempt to allow more intuitive creation. However, many applications have a need for automatic generation, hence the research efforts described in this section.

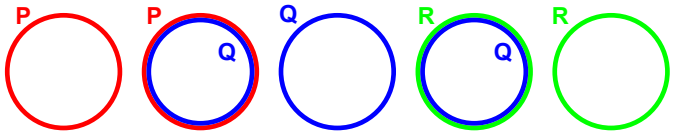

Figure 16: A trivial embedding of the diagram in Figure 11.

There is a trivial solution to this problem that relies on the prodigious use of concurrency and duplicate curve labels. Here, the diagram is drawn as a disconnected set of concurrent circles, one circle for each zone. Each zone has a concurrent circle for each set contained in it. Contours are then a set of disjoint circles. An example of an alternative embedding of the diagram in Figure 11 using this method is shown in Figure 16. It is clear that, whilst adhering to the definition of an Euler diagram, any benefits from visualizing intersection and enclosure that might be gained from using Euler diagrams have been lost. It should also be noted that a similar trivial embedding using concurrency and non-simple curves can be used. Here, circles sharing labels are joined by lines. These lines are then treated as self concurrent curve segments.

Much of the work described in this section takes the approach of dividing the diagram into connected components, which can then be composed in the final drawing. The connected components can be derived from the abstract description in the wellformed case [43] or from the concrete dual in the more general case [22]. A Method to aid layout efficiency by decomposing connected components into subcomponents that are joined by a single curve has been proposed [36].

5.1. Specialized Embedding Methods

Here we look at embedding methods that will draw various classes of abstract description. Sometimes the drawability of an abstract description with a particular method can be derived from the abstract description, others require the embedding of a particular sort of intermediary

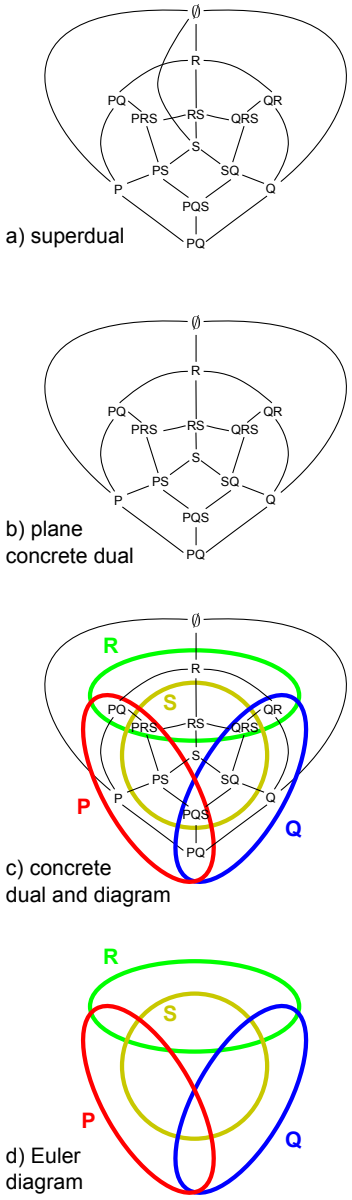

Figure 17: Stages in the embedding of a wellformed diagram using the first embedding method [41]. The abstract description being embedded is $\emptyset$ P Q R S PQ QR.

abstraction (such as a concrete dual) before deciding if the data set can be drawn, yet others can only tell if a diagram is drawable by going through the full embedding process.

The first attempt at a non-trivial embedding was by Flower and Howse [41] with more formalization and other extensions presented in a later paper [40]. This method successfully embeds wellformed diagrams. The method can be summarized as follows:

1. Find the superdual of the abstract description. An example is shown in Figure 17a.
2. Find an embedding of a planar concrete dual, possibly by removing edges, where the connectivity conditions hold (see below) and the face conditions hold (also discussed below), as shown in Figure 17b. If such

a concrete dual cannot be found, then the abstract description cannot be drawn with this method.

3. Draw the curves in the Euler diagram so that they cross concrete dual edges with the same label, and cross each other in faces as prescribed by the face conditions, as shown in Figure 17c.

The above process relies on two concepts introduced by Flower and Howse to check if an Euler diagram can be drawn in a wellformed manner: the *connectivity conditions*; and the *face conditions*. The connectivity conditions relate firstly, to subgraphs of the concrete dual formed from vertices that share a particular curve label and secondly, to subgraphs formed from vertices that do not share a particular curve label. If all such subgraphs for all curve labels are connected, then the diagram can be embedded from the concrete dual with simple closed curves and unique labels, otherwise it cannot, and another concrete dual needs to be found. If no such concrete dual cannot be found, then the diagram cannot be drawn.

The face conditions require the examination of each face in the embedded concrete dual. It is possible to read around the face and, from the 'word' formed, discover which curves cross in the face. If the number and type of crossings meets the face conditions for all faces in the diagram, then the diagram can be embedded from the concrete dual without triple points, otherwise another concrete dual needs to be found, or if no such concrete dual exists, then the diagram cannot be drawn without triple points. In general, exponential time on the number of curves is required to search all possible concrete duals, however, simple heuristics for edge removal can quickly generate wellformed diagrams in a large number of cases.

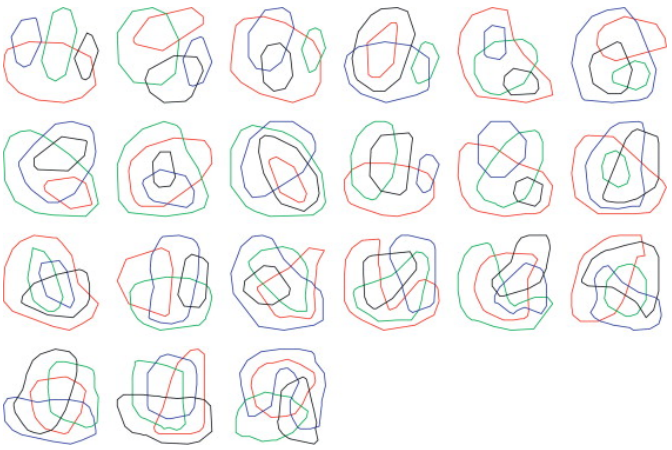

Figure 18: Example outputs from wellformed diagram embedding software [40].

The above insights were encoded in software, which embeds diagrams with up to four curves, see Figure 18 for an example outputs. Later work extended this to arbitrary sized diagrams. This embeds the concrete dual by first taking a triangulation of each face and routing curves through triangles [87].

Work on improving the layout of diagrams embedded using this method attempted to smooth curves, separate contours sufficiently and keep zone/curve sizes reasonably equal has been performed [46], see Figure 19. Alternative layout improvements, inspired by the force directed approach seen in graph drawing [31] has had some success for smaller diagrams [70].

An Euler diagram encoding method allows an inductive embedding for wellformed diagrams [11]. Here curve crossings are labelled with an ordering, however no implementation of the embedding is given. An alternative encoding marking vertices and curve segments indicating inside and outside [28] has been implemented to find the zones in an existing wellformed Euler diagram (taking an embedded diagram and finding its abstract description - the reverse of the generation problem). This task can also also be completed (on any diagram,wellformed or not) by forming the Euler graph and using graph theoretic algorithms to discover the concrete dual, or by using the geometric intersection and containment of the union of regions contained by the curves [26, 121].

Figure 19: Example outputs from software designed to improve the layout of wellformed diagrams [46].

A method to embed any diagrams with up to eight curves was proposed (but no implementation provided) [118]. This relied on constructing a *L_Connected* graph, which is a plane dual which satisfies the connectivity condition that, for every label, the subgraph formed from vertices containing that label are connected. By not insisting on the full connectivity conditions, holes are allowed in contours. Concurrency is also permitted. The paper also gave the result that there are nine curve diagrams that cannot be drawn without using contours consisting of more than one curve or non-simple curves which follows from a result in [67]. The result relies on the observation that there are some diagrams with nine curves that have no planar concrete dual that satisfies the connectivity conditions.

Other work that embeds many abstract descriptions, but not all, includes [94] where intersection graphs (closely related to concrete duals) are laid out in a planar embedding, that is then drawn nicely before the curves are added. The embedding method tends to lead to a great deal of

concurrency, see Figure 10.

Embedding diagrams with desirable shapes, in particular circles has been studied to a some extent. *Pierced diagrams* are a class of Euler diagrams that are drawable with circles [111]. Diagrams can be constructed inductively by using single piercings, where the additional circle intersects a single line segment of an existing circle, and double piercings, where a circle surrounds the intersection point of two existing circles. Single and double piercing can be identified in an abstract description, leading to alternative decompositions sequences of the given abstract description. If a complete decomposition is present, then the abstract description is in the class of pierced diagrams. It should be noted that this class of diagram does not cover all those drawable with circles, for instance, any diagram containing Venn-3 is not in the class of pierced diagrams.

5.2. General Embedding Methods

An important point about the methods discussed in the previous section is that, given an input abstract description, they may not result in an output Euler diagram. Hence the need for a general method that embeds all abstract descriptions. From the result in [67], these methods must allow either non-simple curves or duplicate curve labels.

The first such method relaxed the duplicate curve labels property [85] alongside concurrency, n-points and brushing points, and has been formalized [103]. It extended the first specialized embedding method by Flower and Howse, discussed in Section 5.1. The method finds a concrete dual from the superdual that does not necessarily meet the connectivity or face conditions. The concrete dual has one curve crossing point in each face, so easing the embedding of concurrent edges. Embedding is performed by triangulating each face, choosing one triangle in each face to hold the meeting point. Curves can then be routed through the face to the meeting point. A software implementation of the method includes an attempt at smoothing curves by modifying the layout of the triangulated graph and moving the point of edge/curve intersection to that which evened out the curve angles.

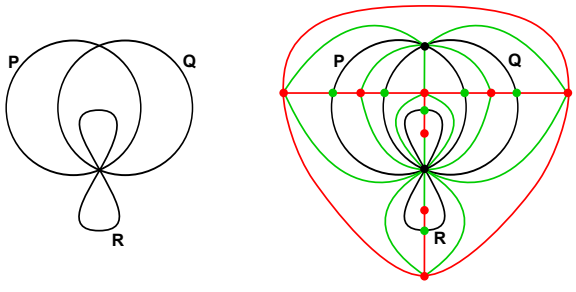

Figure 20: An Euler diagram and its hybrid graph.

A subsequent general embedding technique uses an inductive approach [109] inspired by Venn diagram construction methods [90]. Here, to ensure all abstract descriptions can be drawn, non-simple curves are permitted with concurrency, n-points and brushing points. A decomposition sequence is defined from an abstract description. The curves are then added in sequence. The first curve to be added is drawn as a regular polygon. Subsequent curves are added by finding an appropriate cycle in the *hybrid graph*. This graph is a combination of the concrete dual of the current embedded graph and its Euler graph, see Figure 20. Further edges are added to ensure that all the new zones that are required can, in fact, be added to the diagram. The new curve follows the cycle found in the hybrid graph. The result is a great deal of control over wellformedness, as alternative cycles might add the required zones but have different wellformedness properties. However, the layout quickly becomes convoluted as curves are placed on the diagram. Further work proposes improving the appearance of the diagram by defining an allowed (donut shaped) region though which the curve can pass, and then attempting to fit a familiar shape such as a circle or ellipse in the region [106].

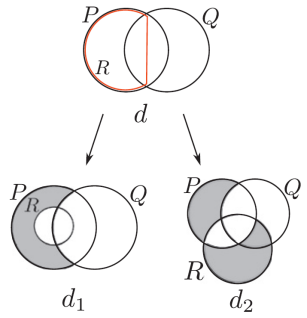

Figure 21: Alternative piercing when adding R [101].

A general embedding method, that extends the piercing method discussed in the last subsection, relies on adding extra zones when piercings cannot be found in the current abstract description [101]. As with Venn diagrams, the extra zones are then shaded to indicate they are empty, see Figure 21. This produces wellformed diagrams, based solely on circles, which can be generated quickly if the minimum number of extra zones is not sought. However, the disadvantage is increased complexity and potential confusion caused by the additional zones. In addition, the radius of the circles reduces rapidly to an unusable size with just a few levels of containment.

5.3. Embedding Open Research Questions

There are areas of current embedding methods that could be greatly improved. Converting a superdual into the best possible concrete dual takes exponential time. Current methods use naïve approaches to find a concrete dual that is often far from the best possible. More sophisticated heuristics for finding an acceptable concrete dual should improve wellformedness in the final diagrams. Another area of general embedding research which has only had limited exploration regards the aesthetics of the final diagrams, where a usable diagram might be one that

maximizes wellformedness. Alternatively (or as well as), the aim may be to draw a diagram with a readable set of curves and zones, so that, for instance, curves are smooth and the areas of zones are not too small or too large. Such improvements to layout, either through optimizing layout through the intermediate embedding steps, or by changing the layout of the final diagram in a post-processing manner, would dramatically increase the usability of the Euler diagrams produced. Of course, it is important to know what a good layout is, and more empirical research needs to be conducted to discover what features of a diagram aid user comprehension.

Dynamic layout of Euler diagrams has had only limited investigation [82] and is a rich area of potential further research. Here, a diagram has an initial layout but the underlying data changes, so requiring that a new version of the diagram is drawn whilst adhering to the previous layout to some extent. This work could draw on research in mental map preservation conducted for dynamic graph drawing, for example discussed in [3, 72]. There is a significant demand for dynamic Euler diagram embedding methods, because, as will be seen later in this paper, dynamic changes to Euler diagrams occur frequently in Euler diagram transformation systems (Section 6) and reasoning systems (Section 7).

There are a number of research questions regarding the embedding of particular shapes. Extending the type of shapes that are used, for example to ellipses or other ovoids, is one area of possible research. Whilst some classes of abstract description (in particular pierced diagrams) are known to be drawable with circles, there are diagrams that can be drawn with circles which do not fit into the current classification. Hence, an interesting research topic is to extend this work to identify more classes of data that can be drawn with circles and to develop embedding techniques for the classes.

6. Transformations

Euler diagram transformations can be defined as algorithmic methods to create a new diagram out of an old one. There is a direct analogy to graph transformation systems, for which there is a widespread and mature research literature [89]. The first Euler diagram transformation systems were developed as part of reasoning systems (see Section 7). For instance, as part of his reasoning system, Hammer introduced three operations: contour addition, contour removal and zone addition, see Figure 24. Given the close analogy with graph transformation systems, it is no surprise to see systems for transforming diagrams that include both Euler diagrams and graphs, for example, spider diagrams (see Section 7), where the transformation rules also include operations modify graphs [54]. As the reasoning systems get more complex, then more complex transformations are required [15]. However, transformations on reasoning systems remain relatively limited as they are tuned to the particular inference rules in the system, and so are somewhat restricted in the type of change that can be applied. In addition, they often leave out the precise details of how to convert between diagrams.

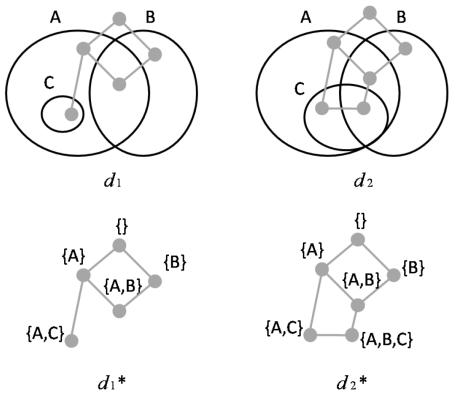

Figure 22: Adding a zone by concrete dual graph transformation [34].

When developing transformation systems which specify the detailed mechanics of how to change a diagram, a question arises as to what representation of the Euler diagram is to be manipulated. When precise curve routing is not a concern, transformations on concrete dual can be used to manipulate an Euler diagram [34]. Here, operations are described to add/remove a curve by adding/removing a subgraph of the concrete dual and to add/remove a zone by adding/removing a single vertex of the concrete dual. Figure 22 shows an operation to add a zone (when read left to right) or remove a zone (when read right to left). In the case shown in Figure 22, adding an edge can remove the brushing point that the zone addition has introduced. The concept of edge manipulation has been generalized with operations on concrete dual edges that change the wellformedness properties of Euler diagrams, whilst retaining the same abstract description [58]. This system does not affect the nodes in the diagram, so it is not possible to use this system to inductively create all possible diagrams.

Transforming the Euler graph [84] has the advantage of working with an existing layout of the diagram, so potentially retaining information about the paths of the curves. Compared with concrete dual edge manipulation, more properties can be varied, in particular, transformations can alter the wellformedness properties in the diagram, altering sections of the diagram that are not wellformed, either increasing the wellformedness in the diagram or swapping between wellformedness properties. For example, the transformation that goes from the second to bottom of Figure 23, where a contour with one non-simple curve is converted to one that has two simple curves. This ensures the wellformedness property of having no non-simple curves is enforced, whilst introducing duplicate curve labels. The operations allowed in this system are: edge contraction/expansion, which removes/creates an edge whilst merging/splitting a vertex (see top to second top in Fig-

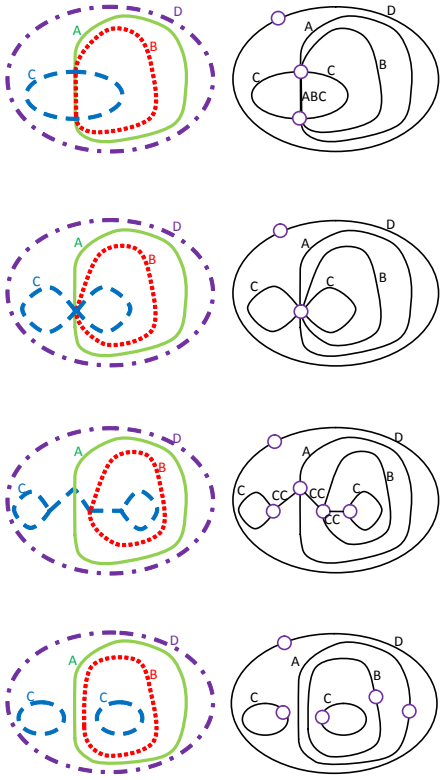

Figure 23: Euler graph transformations [84].

ure 23); edge splitting/merging (which creates/removes a minimal region) and edge deletion/addition (see second bottom to bottom in Figure 23). These transformations can also be used to inductively build diagrams.

6.1. Transformation Open Research Questions

The Euler graph and concrete dual transformation systems described here are currently only conceptual. In order to fully explore their utility, a software implementation would be of great assistance. A software-based transformation tool would allow deeper investigation into which methods are most appropriate to various types of diagram and application. Part of the software system might include heuristics for defining transformation sequences to achieve desired goals.

Transformations on alternative representations of the diagram may also be possible, for example, direct curve manipulation would bring transformation systems closer to manipulating a concrete representation, so allowing the specification of embedding in the transformation. The underlying theory of transformation systems is also lacking. With regard to graph transformation systems, category theory is often used as a mechanism for defining transformations (such as a single or double pushout [89]), and it may be that a similar approach can be taken with Euler diagram transformations.

7. Reasoning Systems

As discussed in previous sections, visual reasoning has been a prime motivation behind the recent research efforts in Euler diagrams. The visual nature of Euler diagrams and their perceived intuitive representation of exclusion, intersection and containment has driven the exploration of Euler diagrams as basis for visual alternatives to symbolic logic notations. A survey by Stapleton [97] detailed the research in this area up to 2004. In this section we summarize these findings, and extend them to the present day.

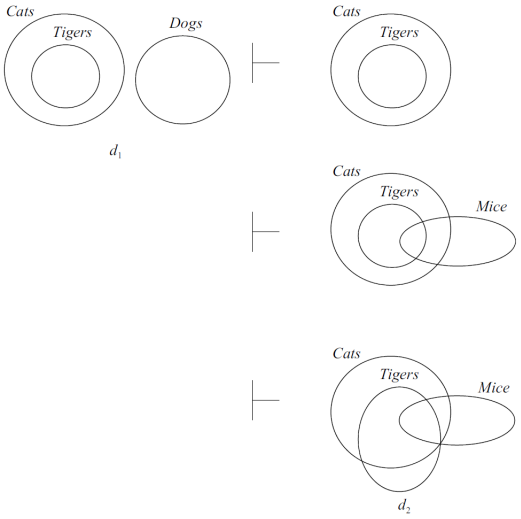

Figure 24: Hammer's reasoning rules from [97].

Hammer introduced an early reasoning using Euler diagrams [51]. Figure 24 demonstrates the use of the three reasoning rules in the system. The first step illustrates the rule of introduction of a new contour, the second step illustrates the of erasure of a contour, and the third step illustrates the rule of weakening (which introduces a zone). Automated theorem proving on Euler diagrams with shading has been implemented in the *Edith* system [105]. This has further rules for adding and removing shaded zones.

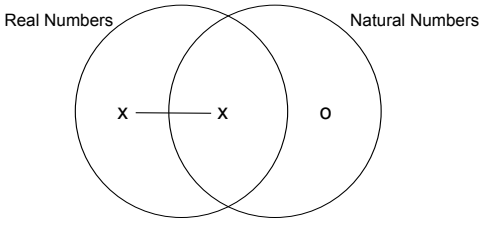

Figure 25: A Venn-Peirce diagram stating that there is at least one real number and all natural numbers are real numbers, from [96].

One of the recurring themes of logical systems using Euler diagrams is that of extending the syntax with graph-like structures so allowing systems to have greater semantics. The original reasoning system by Peirce [76] took this approach. Venn-Peirce diagrams are Venn diagrams where

x is used to represent non-emptiness and o to represent emptiness. The connecting edges represent disjunction. See Figure 25.

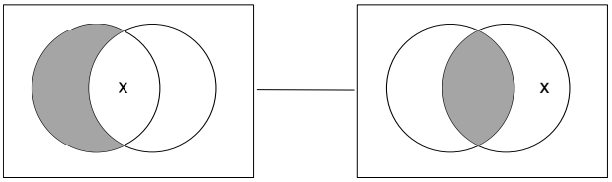

Figure 26: A Venn-II diagram, from [93].

Shin defined Venn-I diagrams [92], which are based on Venn-Peirce diagrams. These revert back to shading to represent empty zones in place of o, as originally used by Venn. However, this is less expressive (as empty zones cannot be included in disjunction). Shin then expands the notation with Venn-II diagrams, where multiple diagrams can be connected by disjunctive lines, and describes ten reasoning rules, six of which are also used for Venn-I diagrams. This work was later extended to include constants [21]. Figure 26 shows a Venn-II diagram.

Euler/Venn diagrams [113] have connected subgraphs which indicate that a particular individual is in one of a group of zones. A directed, acyclic graph representation of the diagrams is used to check the correctness of reasoning steps [114].

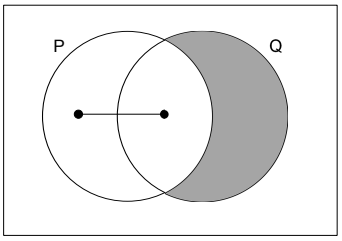

Figure 27: A spider diagram indicating that there is at least one element that is either in zone P or zone PQ and that the zone Q is empty.

Spider diagrams [48, 56, 57, 59] extend Venn-II diagrams, see Figure 27. Here each connected subgraph is called a *spider* and indicates the existence of a distinct element. As with Venn-II, when both spiders and shading are in the same zone, then the diagram is a contradiction. Spider diagrams have been studied extensively by the University of Brighton group responsible for their development and variants have been proved sound and complete [55], as well as equivalent to the a symbolic logic: first order monadic logic with equality [110]. Further work has developed automated theorem provers for spider diagrams [47, 45] and extended the notation to define a second-order diagrammatic logic that has been shown to define a class of star-free regular languages [18].

In an attempt to increase the expressiveness of visual reasoning languages, constraint diagrams were developed [64], see Figure 28 for an example. This represents

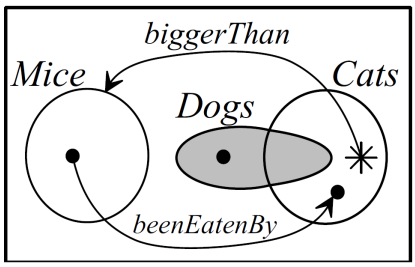

Figure 28: A constraint diagram. From [104].

'No mice are cats or dogs. No dogs are cats. Each cat is bigger than each mouse. There is a mouse that has been eaten by a cat. There is exactly one dog' [104]. Increasing the expressive power of such systems causes problems in identifying the meaning of a diagram due to ambiguity about the order in which items in the diagram should be read, and ambiguity regarding the scope of quantifiers. One solution is to develop an associated reading tree for the diagram [37, 38] which provides a unique semantic interpretation, although providing an ordering can be considered sufficient [17].

Finally in this section, following on from an evaluation of constraint diagrams [98], a diagrammatic system designed to overcome the counterintuitive aspects of the system was developed: Concept Diagrams [19]. They have semantics that are slightly different to constraint diagrams, and permit quantification over sets and relations [102].

7.1. Reasoning Systems Open Research Questions

Diagrammatic reasoning systems are comparatively recent, and so the tools for automatic reasoning such as theorem provers and proof checkers are not as prevalent as tools for textual reasoning systems. One approach to improving access to tools is to translate the diagrammatic statements to text, apply a textual tool, then translate the result back to diagrammatic form. This requires the definition of translation methods, as there may not be obvious one-to-one mappings between all diagrammatic structures and textual logic. It also requires significant study in automatic drawing methods, as the resultant textual statements will be translated to diagrammatic ones that are missing layout information. Alternatively, reasoning tools can be custom-built for the Euler diagram based logics. This has the advantage of tuning the proof sequences and rules for diagrams, However, given the amount of development already devoted to textual reasoning systems, matching the speed of the best textual reasoning systems would need a great deal of research.

The Euler diagram based reasoning systems are not as expressive as the most powerful textual logics and are typically limited to being equivalent to First Order Predicate Logic. This means that many statements expressible in higher order logic simply cannot be expressed in diagrammatic systems. Addressing this issue requires the addition

of higher order concepts to Euler diagram reasoning systems. Such concepts include allowing the representation of powersets and adding inductively defined predicates. Diagrammatic reasoning systems also struggle with representing algebraic expressions, n-ary relations and scoping quantifiers. One approach to dealing with these issues is to develop hybrid (or heterogenous) logics, combining the best of diagrammatic and textual logics. These have the potential to keep the intuitive aspects of diagrammatic reasoning, but to also allow more powerful expressions through the use of text. Whilst some progress has been made on this issue [44, 115], the current state of the art still does not address many of the issues and much more research is still required.

8. Area-Proportional Diagrams

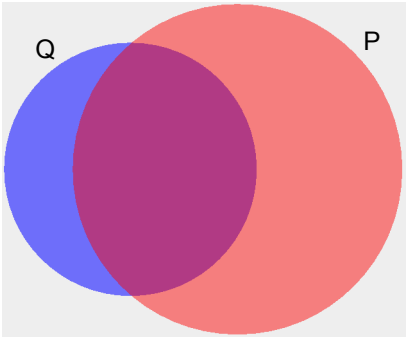

Figure 29: Area-proportional Diagram for $P = 30, Q = 100, PQ = 70$ generated by DrawVenn's 2-circle method [24].

An Euler diagram is *area-proportional* if the areas of the zones in the diagram are considered to be important. As discussed in Section 2, there are a considerable number of application areas with data that is suitable for visualizing using this type of diagram. Typically, the diagram has an area specification associated with it and for each abstract zone this gives a value, which is the desired area of the zone.

More formally, an area-proportional Euler diagram with abstract description (set of abstract zones), Z, also has an associated *area specification*, which is a function $w : Z \to \mathbb{R}^+ \cup \{0\}$. Often the zone outside all curves is not included, as when a diagram is embedded in the plane, this zone must always be infinite.

The values in the area specification are not considered to be absolute, instead, they define the desired relative size of each zone, hence the use of the term 'area-proportional'. For example, an area-proportional Venn-2 shown in Figure 29 represents both the area specification $P = 3, Q = 10, PQ = 7$ and and the area specification $P = 30, Q = 100, PQ = 70$.

If all the areas of the embedded Euler diagram have the relative areas that are given in the area specification, then the diagram is *exact*, otherwise it is *approximate*.

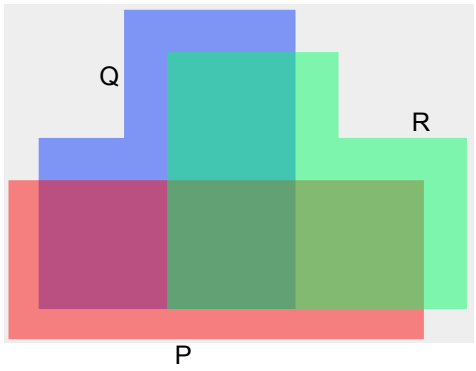

Figure 30: Rectilinear area-proportional diagram generated by DrawVenn's 3-curve method [24]. All zones are the same area.

Much of the early work in generating area-proportional Euler diagrams concentrated on relatively small diagrams. Chow and Ruskey drew exact area-proportional Venn-2 with circles [24] with the DrawVenn software. The radius of the circles is easy to calculate, but a numerical method must be used to find the distance between the centres of the circles, as the equation to calculate the area of intersection is non-invertible. DrawVenn also produces rectilinear area-proportional Venn-3 and those Euler-3 diagrams that include the centre zone, where all three curves intersect. This is achieved by first drawing the centre zone (PQR) as a square, then adding the adjacent zones (PQ, PR and QR) as rectangles. Finally P, Q and R are added. While many of the zones have simple, rectangular shapes, the last three zones to be added may be non-convex. See Figure 30.

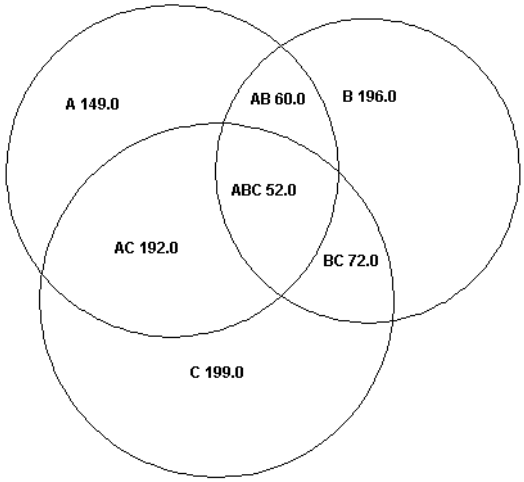

Figure 31: Three circle approximate area-proportional Diagram [23]. The zones are labelled with their desired size.

Further work in three curve layout includes approximate area-proportional Venn-3 circle layout [23], see Figure 31. The position of the circles are first placed with their centres at a distance calculated by treating each three pairs of circles as Venn-2 diagrams, then a hill-climbing search is

applied in an attempt to improve the layout.

VennMaster [66, 65] extends approximate circle area-proportional diagrams to more general cases, see Figure 6. This uses a genetic algorithm search to place the circles (approximated as regular polygons) and, typically, different runs with the same area specification will produce different diagrams. A more statistically sound circle method using the *stress* error measure was then developed by Wilkinson [125] using a faster optimization technique. Both systems may add unwanted zones or remove required zones. This can produce seriously misleading diagrams, and in the case of missing zones, lead to problems in labelling as it is unclear where to annotate a zone that is not on the diagram. Even when the correct zones are present, other problems can occur with approximate diagrams. The diagram shown in Figure 31 demonstrates a potential pitfall: although the zone AB should have larger area than the zone ABC, it clearly has a smaller area, so could lead to misinterpretation on the part of the user.

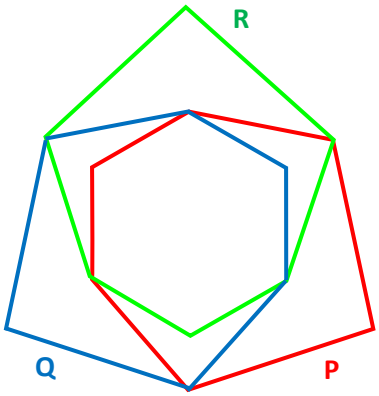

Figure 32: A symmetric exact area-proportional Venn-3 diagram [80].

Restricting the shape of curves often means that an exact diagram cannot be drawn. For example, in general, it is not possible to draw exact diagrams with three circles, and every three circle approach is, by necessity, approximate. In an attempt to address this issue, Venn-3 area-proportional diagrams with ellipses was been explored in the eulerApe system [71], which draws most area specifications exactly. However, it is not possible to draw all three set area specifications with convex curves [24]. This motivated research into methods that use polygons to draw exact diagrams. Such constructions have been developed for Venn-3 [81]. Work giving a classification of all 40 Euler diagrams drawable with 3 curves into those that can be drawn exactly with circles and with convex polygons also provides constructions for the the diagrams [83]. There are also some theoretical results in the symmetric case [80]. A symmetric Venn-3 diagram has $w(P) = w(Q) = W(R)$ and $w(PQ) = w(PR) = W(QR)$, so the drawn diagram can have threefold rotational symmetry, see Figure 32.

It is possible to draw the monotonic class of Euler diagrams with exact area proportions [25] although, often, unnecessary concurrency is present. This builds on work drawing convex intersecting families of simple closed curves [14]. Monotonic diagrams rely on the notion of a *directed dual*, which is a concrete dual of an Euler diagram that has directed edges between two vertices where the curve labels in the target vertex are a proper subset of the curve labels in the source vertex (other edges are left undirected). A *monotonic* Euler diagram has a directed dual with exactly one source and one sink. This implies that the centre zone, that is contained in all curves, must be present in the diagram. Figure 33 shows an Euler diagram that is non-monotonic as there is no zone that is inside all the contours, leading to multiple sources. In addition, both vertices labelled $\emptyset$ and S are sinks.

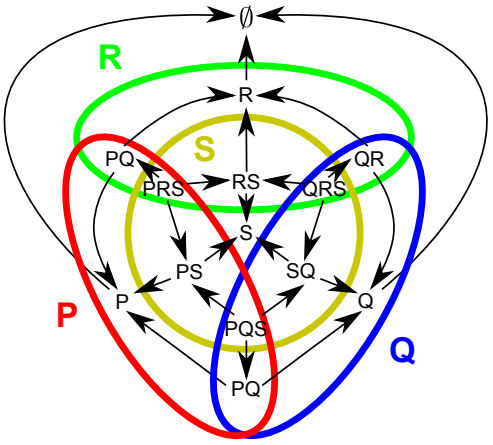

Figure 33: A non-monotonic Euler diagram with its directed dual.

Other publicly available software for drawing area-proportional Euler diagrams includes Vennerable [112] which has developed and integrated a number of techniques for the R statistical software system including those based on circles, triangles and rectangles, many of which are area-proportional. PVENN integrates two and three circle Venn diagrams into *Stata*. Other software systems are aimed at specific application areas, see Section 2.

Finally, a general exact area-proportional Euler diagram drawing method has been proposed [107], but not implemented, based on an inductive embedding approach for diagrams [109] see Section 5.2.

8.1. Area-Proportional Open Research Questions

The work in area proportional diagrams has, up to this date, largely concentrated on specific shapes, particularly rectilinear polygons and circles. As with embedding diagrams that are not area-proportional, there is a question about which shapes are the most effective. The area-proportional case has some interesting differences from the problem of standard Euler diagram layout, as for area-proportional diagrams, the user may be most concerned about being able to compare the size of zones. Hence, making ensuring the shape of zones are amenable to comparison is likely to be more important than the aesthetics

of the overall diagram. For instance, it may be desirable to keep the zone shapes rectangular, at the cost of introducing concurrency in the diagram. Further work is also required to develop a usable system for embedding all possible area specifications, most likely by using arbitrary polygons, rather than specific shapes.

An important question has gone largely unanswered to date, is: how accurate do the zone areas of area-proportional Euler diagrams have to be? Generally, humans cannot precisely gauge the area of regions from sight alone. If some notion of allowable error in zone area could be defined, perhaps by empirical study, then many more area-proportional diagrams might be embedded with specific shapes, with the confidence that the region areas, although not exact, are close enough to the desired values to be useful. It would be also possible to test the error in the diagram, and warn users if the diagram is potentially misleading.

9. Extending Euler Diagrams

In this section we explore some of the syntactic extensions that have been applied to Euler diagrams. In previous sections there has been discussion of Euler diagrams that surround items and Euler diagrams that are drawn with graphs. In addition diagrams drawn in geometries beyond the standard 2D plane have been proposed. Here we explore these types of diagram in more detail.

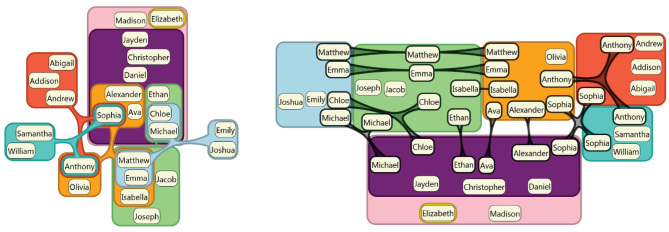

Figure 34: A compact Euler diagram with items on the left. The same data drawn with duplicated items on the right [78].

Euler diagrams are considered a natural way to visualize containment, hence many visualization methods use them to group items (such as visual points, labels or icons) together, see for example, figures 8 and 10.

When automated layout is required, items might be placed by first drawing an Euler diagram and then embedding the items in the correct zones. Two methods given in [78] are shown in Figure 34. The approach in both is to divide the items into a traditional tree hierarchy. In the example on the left, the hierarchy is drawn with rectangles, and the remaining curves added in afterwards. In the example on the right, the items are duplicated, so allowing all rectangles to be placed in the hierarchy, duplicated items are then connected visually. The paper included a study that showed use of duplicated items dramatically improved the accuracy and performance time for most of of the tasks. Placing items can be performed in a random manner or by locating a bounding rectangle in which to add them [69]. Alternatively, a force directed approach where items are repelled both from each other, and from the zone boundaries, can be used to evenly distribute the items (this is used for the vertices in [73]).

Another approach is to take a laid out set of items and then attempt to draw the curves around them. This is less common as it leads to convoluted, hard to follow curves in most cases. Bubble Sets [27] is one such method. It can be used, for example, to show grouping of with geographic data points that are fixed in space, and although the results are not strictly Euler diagrams (extra empty zones can be created) the semantics, that of showing grouping by closed curves, is closely related.

Integrating Euler diagram and graphs is relatively common, in particular, it is common to see notations that include both Euler diagrams and graphs in reasoning systems, see Section 7. Automated drawing of such notations is difficult, as it adds graph layout to the already difficult problem of Euler diagram embedding. One approach is to lay out the Euler diagram and then place the graph in it [73]. Here, after placement of the vertices, a heuristic chooses edge assignment to minimize crossings. This was extended to dynamic diagrams, where the diagram changes due to application of reasoning rules [82]. An alternative approach is to first using graph drawing methods to layout the vertices, followed by the placement of Euler diagram curves around the correct group of vertices using the previously mentioned Bubble Sets method. Intended to ease the manual creation of such diagrams, the sketch recognition of Euler diagrams has been extended to also include graphs [99].

Whilst many Euler diagram visualizations make liberal use of colour for either curves or regions, the concept has been formalized to provide extra dimensions for representing domain features [12].

All the Euler diagrams in this paper up to this point have been assumed to be embedded in the standard Euclidian 2D plane. However, Venn diagram research has often made use of embedding diagrams on the sphere for theoretical purposes [122]. One advantage of embedding on a sphere is that the outside zone that is contained in no curves is no longer infinite. Beyond 2D embeddings, 3D Euler diagrams have been defined as collections of labelled, closed surfaces [79]. Embedding all 3D Euler diagrams can be performed with simple surfaces that have unique labels. This is in contrast to the 2D case where either non-simple curves or duplicate curve labels are required, as discussed in Section 5.

9.1. Extensions Open Research Questions

This section has looked at Euler diagrams drawn with items or graphs, and 3D Euler diagrams. There are there are key research questions to be answered for all these extensions. With regards drawing items in Euler diagrams, there is an issue of ensuring that the zones are large enough to accommodate the number of items that will be drawn

within them. This has a connection with the task of embedding area-proportional Euler diagrams, see Section 8. In particular, the notion of drawing diagrams with approximate zone areas that is given as a future research topic in Section 8.1, which may also suffice for fitting items into zones, as the zones do not have to be a precise area, simply sufficiently large so that the items can be drawn within them.

When using visualizations that include both Euler diagrams and graphs, the current automatic layout methods either employ the strategy of first drawing the Euler diagram, then embedding the graph on it, or use the opposite strategy of first drawing the graph and then embedding the Euler diagram curves to fit with the graph layout. Both of these methods emphasize the layout of one of the structures over the other. In most cases it would be more preferable to integrate the drawing, so compromising the layout of both Euler diagram and graph for a better overall diagram. Hence, the development of methods to draw both structures simultaneously should result in significantly more usable visualizations.

In terms of 3D Euler diagrams, the first (and, currently, only) paper in the area [79] outlines a number of research questions that could be addressed. These include questions that have been answered for the 2D case, such as the wellformedness properties that must be broken when embedding 3D Euler diagrams and what shapes can be used for various data sets.

10. Conclusions

In this paper, we have surveyed research on Euler diagrams. They are widely used because their graphical representation of containment, intersection and exclusion is considered an effective way to visualize data. Recent research efforts into reasoning systems based on Euler diagrams, their embedding and transformation, as well as the development of area-proportional systems has advanced the state-of-the art considerably in recent years. This work has meant that Euler diagrams are beginning to be adopted where previously their use was not feasible.

In this paper, potential future work has been given at the end of each section. The wider application of Euler diagrams will largely rely on the development of better techniques for their comprehensible layout, particularly when items or graphs are included in the visualization. Much of the key research effort in the future will relate to discovering new embedding techniques, alongside empirical work that informs the algorithm designer about what are the best aesthetic considerations to take into account for assisting users in getting the most out of Euler diagrams.

Acknowledgements Peter Rodgers is supported by a Royal Society Industrial Fellowship.

References

[1] A.S. Al-Watban and Z.R. Yang. Bimodal gene prediction via gap maximisation. In *BIOCOMP'12*, 2012.

[2] B. Alper, N. Riche, G. Ramos, and M. Czerwinski. Design study of linesets, a novel set visualization technique. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2259–2267, 2011.

[3] D. Archambault, H. Purchase, and B. Pinaud. Animation, small multiples, and the effect of mental map preservation in dynamic graphs. *Visualization and Computer Graphics, IEEE Transactions on*, 17(4):539–552, 2011.

[4] M.E. Baron. A note on the historical development of logic diagrams: Leibniz, euler and venn. *The mathematical gazette*, 53(384):113–125, 1969.

[5] J. Barwise and A. Shimojima. Surrogate reasoning. *Cognitive Studies: Bulletin of the Japanese Cognitive Science Society*, 2(4):7–27, 1995.

[6] G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing: Algorithms for the visualization of graphs*. Prentice Hall, 1999.

[7] F. Benoy and P. Rodgers. Evaluating the comprehension of Euler diagrams. In *11th International Conference on Information Visualization*, pages 771–778. IEEE Computer Society, 2007.

[8] F. Bertault and P. Eades. Drawing hypergraphs in the subset standard. In *Proceedings of the 8th International Symposium on Graph Drawing*, volume 1984 of *LNCS*, pages 164 – 169. Springer Verlag, 2000.

[9] A. Blake, G. Stapleton, P. Rodgers, L. Cheek, and J. Howse. Does the orientation of an euler diagram affect user comprehension? In *Proceedings of the 18th International Conference on Distributed Multimedia Systems, International Workshop on Visual Languages and Computing (VLC 2012)*, volume 18, pages 185–190. Knowledge Systems Institute, 2012.

[10] P. Bottoni, G. Cordasco, R. De Chiara, A. Fish, and V. Scarano. Personalised resource categorisation using euler diagrams. In *End-User Development: Third International Symposium, IS-EUD 2011, Torre Canne, Italy, June 7-10, 2011, Proceedings*, volume 6654, page 251. Springer, 2011.

[11] P. Bottoni, G. Costagliola, and A. Fish. Euler diagram encodings. In *Diagrams*, pages 148–162, 2012.

[12] P. Bottoni and A. Fish. Coloured euler diagrams: a tool for visualizing dynamic systems and structured information. *Diagrammatic Representation and Inference*, pages 39–53, 2010.

[13] T.E. Brand and A.J. Sherlock. Vk'+ v'k. venn diagrams or karnaugh maps, but not both? *Mathematics in School*, 1(7):4–7, 1972.

[14] B. Bultena, B. Grünbaum, B.G. Unbaum, and F. Ruskey. Convex drawings of intersecting families of simple closed curves. In *11th Canadian Conference on Computational Geometry*, pages 18–21, 1999.

[15] J. Burton, G. Stapleton, and A. Hamie. Transforming constraint diagrams. *Visual Languages and Logic*, pages 62–80, 2009.

[16] P.C. Carvalho, J.R. Yates III, and V.C. Barbosa. Analyzing shotgun proteomic data with Patternlab for proteomics. *Current Protocols in Bioinformatics*, pages 13–13, 2010.

[17] P. Chapman and G. Stapleton. Creating a second order diagrammatic logic. *Diagrammatic Representation and Inference*, pages 298–300, 2010.

[18] P. Chapman and G. Stapleton. Introducing second-order spider diagrams for defining regular languages. In *VL/HCC*, pages 159–167, 2010.

[19] P. Chapman, G. Stapleton, J. Howse, and I. Oliver. Deriving sound inference rules for concept diagrams. In *Visual Languages and Human-Centric Computing (VL/HCC), 2011 IEEE Symposium on*, pages 87–94. IEEE, 2011.

[20] H. Chen and P.C. Boutros. Venndiagram: a package for the generation of highly-customizable venn and euler diagrams in r. *BMC bioinformatics*, 12(1):35, 2011.

[21] L. Choudhury and M. K. Chakraborty. On extending Venn diagrams by augmenting names of individuals. In *Proceedings of*

3rd International Conference on the Theory and Application of Diagrams, volume 2980 of *LNAI*, pages 142–146. Springer-Verlag, March 2004.

[22] S. Chow. *Generating and Drawing Area-Proportional Euler and Venn Diagrams*. PhD thesis, University of Victoria, 2007.

[23] S. Chow and P. Rodgers. Constructing area-proportional Venn and Euler diagrams with three circles. In *Proceedings of Euler Diagrams 2005*, 2005.

[24] S. Chow and F. Ruskey. Drawing area-proportional Venn and Euler diagrams. In *Proceedings of Graph Drawing 2003, Perugia, Italy*, volume 2912 of *LNCS*, pages 466–477. Springer-Verlag, 2003.

[25] S. Chow and F. Ruskey. Towards a general solution to drawing area-proportional Euler diagrams. In *Euler Diagrams 2004*, volume 134 of *ENTCS*, pages 3–18. ENTCS, 2005.

[26] R. Clark. Fast zone discrimination. In *Proceedings of the VLL 2007 workshop on Visual Languages and Logic*, pages 41–54. Citeseer, 2007.

[27] C. Collins, G. Penn, and S. Carpendale. Bubble sets: Revealing set relations with isocontours over existing visualizations. *Visualization and Computer Graphics, IEEE Transactions on*, 15(6):1009–1016, 2009.

[28] G. Cordasco, R. De Chiara, and A. Fish. Efficient on-line algorithms for euler diagram region computation. *Comput. Geom.*, 44(1):52–68, 2011.

[29] R. DeChiara, U. Erra, and V. Scarano. VennFS: A Venn diagram file manager. In *Proceedings of Information Visualisation*, pages 120–126. IEEE Computer Society, 2003.

[30] D. Dorling. Cartograms for visualizing human geography. *Visualization in Geographical Information Systems. Wiley*, 85:102, 1994.

[31] P. Eades. A heuristic for graph drawing. *Congressus Numerantium*, 42:149–160, 1984.

[32] L. Euler. Lettres à une Princesse d'Allemagne sur divers sujets de physique et de philosophie. *Letters*, 2:102–108, 1775.

[33] M. Farfel, L. DiGrande, R. Brackbill, A. Prann, J. Cone, R. Samuel R, D. Friedman, D. Walker, G. Pezeshki, P. Thomas, Sandro Galea, D. Williamson, T. Frieden, and L. Thorpe. An overview of 9/11 experiences and respiratory and mental health conditions among world trade center health registry enrollees. *Journal of Urban Health*, 85(6):880–909, 2008.

[34] A. Fish. Euler diagram transformations. *Electronic Communications of the EASST*, 18, 2009.

[35] A. Fish and J. Flower. Abstractions of Euler diagrams. In *Euler Diagrams 2004*, volume 134 of *ENTCS*, pages 77–101. ENTCS, 2005.

[36] A. Fish and J. Flower. Euler diagram decomposition. In *Diagrams 2008*. Springer, 2008.

[37] A. Fish, J. Flower, and J. Howse. A reading algorithm for constraint diagrams. In *IEEE Symposium on Human Centric Computing Languages and Environments, Auckland, New Zealand*, pages 161–168. IEEE, September 2003.

[38] A. Fish and J. Howse. Computing reading trees for constraint diagrams. In *AGTIVE '03, Applications of Graph Transformations with Industrial Relevance, Charlottesville, Virginia, September*, pages 260–274. Springer-Verlag, 2003.

[39] A. Fish, B. Khazaei, and C. Roast. User-comprehension of euler diagrams. *Journal of Visual Languages and Computing*, 22(5):340 – 354, 2011.

[40] J. Flower, A. Fish, and J. Howse. Euler diagram generation. *Journal of Visual Languages and Computing*, 19:675–694, 2008.

[41] J. Flower and J. Howse. Generating Euler diagrams. In *Proceedings of 2nd International Conference on the Theory and Application of Diagrams*, pages 61–75, Georgia, USA, 2002. Springer.

[42] J. Flower, J. Howse, and J. Taylor. Nesting in Euler diagrams. In *International Workshop on Graph Transformation and Visual Modeling Techniques*, pages 99–108, 2002.

[43] J. Flower, J. Howse, and J. Taylor. Nesting in Euler diagrams: syntax, semantics and construction. *Software and Systems Modelling*, 3:55–67, 2004.

[44] J. Flower, J. Howse, J. Taylor, and S. Kent. A visual framework for modelling with heterogeneous notations. In *Proceedings of Human Centric Computing Languages and Environments*, pages 71–73. IEEE, 2002.

[45] J. Flower, J. Masthoff, and G. Stapleton. Generating readable proofs: A heuristic approach to theorem proving with spider diagrams. In *Proceedings of 3rd International Conference on the Theory and Application of Diagrams*, volume 2980 of *LNAI*, pages 166–181, Cambridge, UK, 2004. Springer.

[46] J. Flower, P. Rodgers, and P. Mutton. Layout metrics for Euler diagrams. In *7th International Conference on Information Visualisation*, pages 272–280. IEEE Computer Society Press, 2003.

[47] J. Flower and G. Stapleton. Automated theorem proving with spider diagrams. In *Proceedings of Computing: The Australasian Theory Symposium*, volume 91 of *ENTCS*, pages 116–132, Dunedin, New Zealand, January 2004. Science Direct.

[48] J. Gil, J. Howse, and S. Kent. Formalising spider diagrams. In *Proceedings of IEEE Symposium on Visual Languages (VL99), Tokyo*, pages 130–137. IEEE Computer Society Press, September 1999.

[49] C. Gurr and K. Tourlas. Towards the principled design of software engineering diagrams. In *Proceedings of 22nd International Conference on Software Engineering*, pages 509–518. ACM Press, 2000.

[50] H. Hagy. Indexed, November 2012.

[51] E. Hammer. *Logic and Visual Information*. CSLI Publications, 1995.

[52] A. Harwood-Nuss. *The clinical practice of emergency medicine*. Lippincott Williams & Wilkins, 2001.

[53] P. Hayes, T.C. Eskridge, R. Saavedra, T. Reichherzer, M. Mehrotra, and D. Bobrovnikoff. Collaborative knowledge capture in ontologies. In *Proceedings of the 3rd international conference on Knowledge capture*, K-CAP '05, pages 99–106, New York, NY, USA, 2005. ACM.

[54] J. Howse, F. Molina, and J. Taylor. On the completeness and expressiveness of spider diagram systems. In *Proceedings of 1st International Conference on the Theory and Application of Diagrams*, pages 26–41, Edinburgh, UK, September 2000. Springer.

[55] J. Howse, F. Molina, and J. Taylor. A sound and complete diagrammatic reasoning system. In *Proceedings. ASC 2000: 3rd IASTED International Conference on Artificial Intelligence and Soft Computing*, pages 402–408, Banff, 2000. IASTED/ACTA Press.

[56] J. Howse, F. Molina, J. Taylor, and S. Kent. Reasoning with spider diagrams. In *Proceedings of IEEE Symposium on Visual Languages (VL99), Tokyo*, pages 138–147. IEEE Computer Society Press, September.

[57] J. Howse, F. Molina, J. Taylor, S. Kent, and J. Gil. Spider diagrams: A diagrammatic reasoning system. *Journal of Visual Languages and Computing*, 12(3):299–324, June 2001.

[58] J. Howse, P. Rodgers, and G. Stapleton. Changing Euler Diagram Properties by Edge Transformation of Euler Dual Graphs. In *IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC 2009)*, volume 25, pages 177–184. IEEE, September 2009.

[59] J. Howse, G. Stapleton, and J. Taylor. Spider diagrams. *LMS Journal of Computation and Mathematics*, 8:145–194, 2005.

[60] J. Howse, G. Stapleton, K. Taylor, and P. Chapman. Visualizing ontologies: a case study. In *Proceedings of the 10th international conference on The semantic web-Volume Part I*, pages 257–272. Springer-Verlag, 2011.

[61] T. Hulsen, J. De Vlieg, and W. Alkema. Biovenn–a web application for the comparison and visualization of biological lists using area-proportional venn diagrams. *BMC genomics*, 9(1):488, 2008.

[62] E. Ip. Visualizing multiple regression. *Journal of Statistics*

Education, 9(1), 2001.

[63] C. John, A. Fish, J. Howse, and J. Taylor. Exploring the notion of clutter in Euler diagrams. In *4th International Conference on the Theory and Application of Diagrams*, pages 267–282, Stanford, USA, 2006. Springer.

[64] S. Kent. Constraint diagrams: Visualizing invariants in object oriented modelling. In *Proceedings of OOPSLA97*, pages 327–341. ACM Press, October 1997.

[65] H. Kestler, A. Muller, T. Gress, and M. Buchholz. Generalized Venn diagrams: A new method for visualizing complex genetic set relations. *Bioinformatics*, 21(8):1592–1595, 2005.

[66] H. Kestler, A. Muller, J. Kraus, M. Buchholz, T. Gress, H. Liu, D. Kane, B. Zeeberg, and J. Weinstein. Vennmaster: Area-proportional Euler diagrams for functional GO analysis of microarrays. *BMC Bioinformatics*, 9(67), 2008.

[67] O. Lemon and I. Pratt. Spatial logic and the complexity of diagrammatic reasoning. *Machine GRAPHICS and VISION*, 6(1):89–108, 1997.

[68] E.M. Luks. Hypergraph isomorphism and structural equivalence of boolean functions. In *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, STOC '99, pages 652–658, New York, NY, USA, 1999. ACM.

[69] L. Micallef, P. Dragicevic, and J.-D. Fekete. Assessing the effect of visualizations on bayesian reasoning through crowdsourcing. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12):2536 –2545, dec. 2012.

[70] L. Micallef and P. Rodgers. Poster: Force-directed layout for Euler diagrams. In *Compendium of IEEE Information Visualization 2009 (InfoVis'09)*, volume 15, Atlantic City, New Jersey, USA, October 2009. IEEE Computer Society, IEEE.

[71] L. Micallef and P. Rodgers. Poster: Drawing area-proportional venn-3 diagrams using ellipses. In *ACM Student Research Competition and Poster Session*, Grace Hopper Celebration of Women in Computing, 2012.

[72] K. Misue, P. Eades, W. Lai, and K. Sugiyama. Layout adjustment and the mental map. *Journal of Visual Languages and Computing*, 6(2):183–210, June 1995.

[73] P. Mutton, P. Rodgers, and J. Flower. Drawing graphs in Euler diagrams. In *Proceedings of 3rd International Conference on the Theory and Application of Diagrams*, volume 2980 of *LNAI*, pages 66–81, Cambridge, UK, March. Springer.

[74] I. Oliver, J. Howse, G. Stapleton, E. Nuutila, and S. Törmä. Visualizing and specifying ontologies using diagrammatic logics. In *5th Australasian Ontologies Workshop*, volume 112 of *CRPIT*, pages 87–104, 2009.

[75] OMG. UML 2.0 specification. Available from http://www.omg.org, 2004.

[76] C. Peirce. *Collected Papers*, volume 4. Harvard University Press, 1933.

[77] M. Pirooznia, V. Nagarajan, Y. Deng, et al. GeneVenn-a web application for comparing gene lists using venn diagrams. *Bioinformation*, 1(10):420–422, 2007.

[78] N. Riche and T. Dwyer. Untangling Euler diagrams. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1090–1099, 2010.

[79] P. Rodgers, J. Flower, and G. Stapleton. Introducing 3D Venn and Euler diagrams. In P. Chapman and L. Micallef, editors, *Proceedings of the 3rd International Workshop on Euler Diagrams 2012*, volume 854 of *CEUR-WS*, pages 92–106. CEUR-WS.org, July 2012.

[80] P. Rodgers, J. Flower, G. Stapleton, and J. Howse. Some results for drawing area proportional Venn3 with convex curves. In *Information Visualization*, pages 667–672. IEEE, 2009.

[81] P. Rodgers, J. Flower, G. Stapleton, and J. Howse. Drawing Area-Proportional Venn-3 Diagrams with Convex Polygons. In *Diagrams 2010*, LNCS (LNAI) 6170, pages 54–68. Springer, August 2010.

[82] P. Rodgers, P. Mutton, and J. Flower. Dynamic Euler diagram drawing. In *Visual Languages and Human Centric Computing, Rome, Italy*, pages 147–156. IEEE Computer Society Press, September 2004.

[83] P. Rodgers, G. Stapleton, J. Flower, and J. Howse. Drawing area-proportional Euler diagrams representing up to three sets. *Visualization and Computer Graphics, IEEE Transactions on*, 2013.

[84] P. Rodgers, G. Stapleton, J. Howse, and L. Zhang. Euler Graph Transformations for Euler Diagram Layout. In Christopher Hundhausen, Emmanuel Pietriga, Paloma Diaz, and Mary Beth Rosson, editors, *IEEE Symposium on Visual Languages and Human Centric Computing 2010*, pages 111–118. IEEE, IEEE, September 2010.

[85] P. Rodgers, L. Zhang, and A. Fish. General Euler diagram generation. In *5th International Conference on the Theory and Application of Diagrams*, pages 13–27. Springer, 2008.

[86] P. Rodgers, L. Zhang, and H. Purchase. Wellformedness properties in Euler diagrams: Which should be used? *Transactions on Visualization and Computer Graphics*, 18(7):1089–1100, July 2012.

[87] P. Rodgers, L. Zhang, G. Stapleton, and A. Fish. Embedding wellformed Euler diagrams. In *12th International Conference on Information Visualization*, pages 585–593. IEEE, 2008.

[88] P. Rongdeep and B. Kalita. Properties of some euler graphs constructed from euler diagram. *International Journal of Applied Science and Engineering Research*, 1(2):232–237, 2012.

[89] G. Rozenberg, editor. *Handbook of Graph Grammars and Computing by Graph Transformations, Volume 1: Foundations*. World Scientific, 1997.

[90] F. Ruskey. A survey of Venn diagrams. *Electronic Journal of Combinatorics*, 1997. www.combinatorics.org/Surveys/ds5/VennEJC.html.

[91] Y. Sato, K. Mineshima, and R. Takemura. The efficacy of euler and venn diagrams in deductive reasoning: Empirical findings. In *Diagrammatic Representation and Inference: 6th International Conference, Diagrams 2010, Portland, OR, USA, August 9-11, 2010, Proceedings*, volume 6170, page 6. Springer, 2010.

[92] S.-J. Shin. *The Logical Status of Diagrams*. Cambridge University Press, 1994.

[93] S.-J. Shin and O. Lemon. Stanford enyclopedia of philosophy: Diagrams. http://plato.stanford.edu/entries/diagrams/, 2008.

[94] P. Simonetto, D. Auber, and D. Archambault. Fully automatic visualisation of overlapping sets. *Computer Graphics Forum*, 28(3):967–974, 2009.

[95] J. Soriano, K. Davis B. Coleman, G. Visick, D. Mannino, and N. Pride. The proportional Venn diagram of obstructive lung disease. *Chest*, 124:474–481, 2003.

[96] G. Stapleton. *Reasoning with Constraint Diagrams*. PhD thesis, University of Brighton, August 2004.

[97] G. Stapleton. A survey of reasoning systems based on Euler diagrams. In *Proceedings of Euler Diagrams 2004, Brighton, UK*, volume 134 of *ENTCS*, pages 127–151, 2005.

[98] G. Stapleton and A. Delaney. Evaluating and generalizing constraint diagrams. *Journal of Visual Languages and Computing*, 19(4):499–521, 2008.

[99] G. Stapleton, A. Delaney, P. Rodgers, and B. Plimmer. Recognising sketches of Euler diagrams augmented with graphs. In *2011 International Workshop on Visual Languages and Computing*, volume 17 of *International Conference on Distributed Multimedia Systems*, pages 279–284, August 2011.

[100] G. Stapleton, A. Fish, and P. Rodgers. Abstract Euler diagram isomorphism. In *Visual Languages and Computing*. Knowledge Systems Institute, 2008.

[101] G. Stapleton, J. Flower, P. Rodgers, and J. Howse. Automatically drawing Euler diagrams with circles. *J. Vis. Lang. Comput.*, 23(3):163–193, June 2012.

[102] G. Stapleton, J. Howse, P. Chapman, I. Oliver, and A. Delaney. What can concept diagrams say? In *Diagrams*, pages 291–293, 2012.

[103] G. Stapleton, J. Howse, and P. Rodgers. A graph theoretic approach to general Euler diagram drawing. *Theoretical Computer Science*, 441(1):91–112, 2010.

[104] G. Stapleton, J. Howse, and J. Taylor. A constraint diagram

reasoning system. In *Proceedings of International Conference on Visual Languages and Computing*, pages 263–270. Knowledge Systems Insitute, 2003.

[105] G. Stapleton, J. Masthoff, J. Flower, A. Fish, and J. Southern. Automated theorem proving in Euler diagrams systems. *Journal of Automated Reasoning*, 39:431–470, 2007.

[106] G. Stapleton and P. Rodgers. Drawing Euler diagrams with circles and ellipses. In *VL/HCC*, pages 209–212, 2011.

[107] G. Stapleton, P. Rodgers, and J. Howse. A general method for drawing area-proportional euler diagrams. *Journal of Visual Languages and Computing*, 22(6):426–442, December 2011. In press, available online.

[108] G. Stapleton, P. Rodgers, J. Howse, and J. Taylor. Properties of Euler diagrams. In *Proceedings of Layout of Software Engineering Diagrams*, pages 2–16. EASST, 2007.

[109] G. Stapleton, P. Rodgers, J. Howse, and L. Zhang. Inductively generating Euler diagrams. *accepted for IEEE Transactions on Visualization and Computer Graphics*, 2009.

[110] G. Stapleton, S. Thompson, J. Howse, and J. Taylor. The expressiveness of spider diagrams. *Journal of Logic and Computation*, 14(6):857–880, December 2004.

[111] G. Stapleton, L. Zhang, J. Howse, and P. Rodgers. Drawing Euler diagrams with circles: The theory of piercings. *Visualization and Computer Graphics, IEEE Transactions on*, 17(7):1020–1032, 2011.

[112] J. Swinton. Venn diagrams with the vennerable package. 2007.

[113] N. Swoboda. Implementing Euler/Venn reasoning systems. In M. Anderson, B. Meyer, and P. Olivier, editors, *Diagrammatic Representation and Reasoning*, pages 371–386. Springer-Verlag, 2001.

[114] N. Swoboda and G. Allwein. Using DAG transformations to verify Euler/Venn homogeneous and Euler/Venn FOL heterogeneous rules of interence. In *Proceedings of GT-VMT*, ENTCS. Elsevier Science, 2002.

[115] N. Swoboda and G. Allwein. Heterogeneous reasoning with Euler/Venn diagrams containing named constants and FOL. In *Proceedings of Euler Diagrams 2004*, volume 134 of *ENTCS*. Elsevier Science, 2005.

[116] A. Treisman and J. Souther. Search asymmetry: a diagnostic for preattentive processing of separable features. *Journal of Experimental Psychology: General; Journal of Experimental Psychology: General*, 114(3):285, 1985.

[117] J. Venn. On the diagrammatic and mechanical representation of propositions and reasonings. *Phil.Mag*, 1880.

[118] A. Verroust and M.-L. Viaud. Ensuring the drawability of Euler diagrams for up to eight sets. In *Proceedings of 3rd International Conference on the Theory and Application of Diagrams*, volume 2980 of *LNAI*, pages 128–141, Cambridge, UK, 2004. Springer.

[119] J. Vestbo. Copd, diagrams and traditions: time to move on? *Thorax*, 63(9):755–756, 2008.

[120] M. Wang, B. Plimmer, P. Schmieder, G. Stapleton, P. Rodgers, and A. Delaney. SketchSet: creating Euler diagrams using pen or mouse. In *2011 IEEE Symposium on Visual Languages and Human Centric Computing (VL/HCC 2011)*, pages 75–82. IEEE, September 2011.

[121] M. Wang, B. Plimmer, P. Schmieder, G. Stapleton, P. Rodgers, and A. Delaney. Sketchset: Creating euler diagrams using pen or mouse. In *IEEE Symposium on Visual Languages and Computing*, pages 75–82. IEEE, 2011.

[122] M.R.N. Weston. *Symmetries of Venn diagrams on the sphere*. PhD thesis, University of Victoria, 2009.

[123] Wikipedia. Euler diagram of quadrilateral types, November 2012.

[124] Wikipedia. Supernational bodies, November 2012.

[125] L. Wilkinson. Exact and approximate area-proportional circular venn and euler diagrams. *Visualization and Computer Graphics, IEEE Transactions on*, 18(2):321–331, 2012.

[126] B.R. Zeeberg, W. Feng, G. Wang, M.D. Wang, A.T. Fojo, M. Sunshine, S. Narasimhan, D.W. Kane, W.C. Reinhold, S. Lababidi, et al. Gominer: a resource for biological interpretation of genomic and proteomic data. *Genome Biol*, 4(4):R28, 2003.